INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Model Reduction and Adaption of Optimum-shape design in aerodynamics by Neural Networks

Marios K. Karakasis — Jean-Antoine Désidéri

N° 4503

10 July 2002

THÈME 4

ISSN 0249-6399 ISRN INRIA/RR--4503--FR+ENG

INRIA
SOPHIA ANTIPOLIS

Model Reduction and Adaption of Optimum-shape design in aerodynamics by Neural Networks

Marios K. Karakasis , Jean-Antoine Désidéri

Thème 4 — Simulation et optimisation
de systèmes complexes
Projet Opale

Rapport de recherche n° 4503 — 10 July 2002 — 28 pages

Abstract: A method to reduce the dimension of the initial search space in an optimization problem is proposed. The method consists in the identification of the sub-space with the greatest impact on the cost or fitness function. Optimization is restricted in this sub-space, achieving, thus, a considerable reduction of the computational cost, due to more effective exploration. The Model Reduction is the result of mathematical analysis performed on approximations of the cost/fitness function supplied by Artificial Neural Networks, trained during the optimization process. The Model Reduction is coupled with Genetic Algorithms and performed in a self-adaptive way during the genetic evolution.

Key-words: Optimization, Model Reduction, Artificial Neural Networks, Genetic Algorithms

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex (France)
Téléphone : +33 4 92 38 77 77 — Télécopie : +33 4 92 38 77 65

Réduction de modèle et adaptation en conception optimale de forme aérodynamique par réseaux de neurones

Résumé : On propose une méthode pour réduire la dimension de l'espace de recherche dans un problème d'optimisation. La méthode consiste à identifier le sous-espace engendré par les vecteurs propres de la matrice hessienne qui ont le plus d'influence sur la réduction de la fonctionnelle coût. Quand l'optimisation est limitée à un tel sous-espace, des réductions substantielles du coût de calcul sont observées grâce à une exploration plus efficace. La réduction de modèle est faite par l'analyse mathématique d'approximations de la fonctionnelle coût fournie par réseaux de neurones, dont l'apprentissage se fait par le processus d'optimisation initial. La réduction de modèle est couplée à des algorithmes génétiques de manière auto-adaptative tout au long du processus d'évolution.

Mots-clés : Optimisation, Réduction de Modèle, Réseaux de Neurones, Algorithmes Génétiques

Contents

List of Tables

List of Figures

1 Introduction

The starting point of this work has been a, quite obvious, observation: in an optimization problem not all of the variables are of the same importance, or, having advanced in the optimization process, no more all of the design parameters are important. Therefore, it would be more efficient to focus on the directions of the search space with the highest "pay-off", instead of consuming CPU time to explore directions with a minor impact on the fitness or cost function value.

This is particularly important in Computational Fluid Dynamics (*CFD*) inverse design or optimization problems, where a high number of design variables is a common situation. In addition, the complexity of these problems, resulting in a multimodal cost function, makes the robustness of Evolutionary Algorithms outweigh their relatively high computational cost. Hence, the adaption of the search space to the morphology of the optimization problem has been coupled with a Genetic Algorithm (*GA*, [1], [2]), the latter being, perhaps, the most widely used amidst the stochastic optimization methods.

The reduction of the model, as a result of the adaption of the search space, requires a number of mathematical computations performed on the fitness function. However, quite seldom is the latter available in a closed form. In most cases, its value is derived from a complex analysis on a candidate solution — a typical sequence in *CFD* problems would be: shape generation, grid generation, flow field computation, post-processing of the results and calculation of the cost function.

To circumvent this inconvenience a surrogate model should be used, offering a reliable but not expensive approximation to the costly and not available in closed form fitness function. As such Artificial Neural Networks (*ANNs*) have been used and especially Radial Basis Function (*RBF*) networks. The choice of this type of network has been guided by its low cost of training — in conjunction with the context in which they are used — combined with satisfactory fidelity and by the extensive experience on this type of neural networks ([3], [4], [5]).

Throughout a number of test-cases it will be demonstrated the reduction of the optimization cost, which is achieved by adapting the search space to the optimization problem itself. However, this is strongly related to the quality of the approximations that yield the employed *ANNs*.

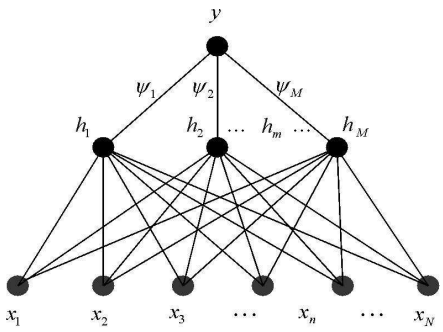

Figure 1: A typical *RBF* network with a single output unit.

2 Radial Basis Function Networks

A typical *RBF* network is illustrated in Fig. 1. It consists of a single layer of M hidden nodes, between the N input nodes and the output. By presenting the network with the N design variables — which in the present applications correspond to the parameters defining an aerodynamic shape —, signals propagate in the forward direction with computations carried over the hidden units and the network response appears at the output node. The signal propagation involves a non-linear mapping, $H : \mathbb{R}^N \to \mathbb{R}^M$, to the hidden nodes space, which is followed by the linear mapping $\Psi : \mathbb{R}^M \to \mathbb{R}$ to the output. In an *RBF* network the only adjustable parameters are the M synaptic weights $(\psi_1, \psi_2, \ldots, \psi_M) = \psi$ associated with the links that perform the linear mapping to the output node value (see also [6], [7],[8]).

Each hidden node is associated with a point c in the space $\mathbb{R}^N$ of design variables, which is called the center of the node. The non-linear mapping H is performed by the application, on every hidden node, of an activation function F_m, $m = 1, M$ to the deviation of the input $x \in \mathbb{R}^N$ from the corresponding center. Thus:

$$H_m(x) = F_m(\|x - c_m\|) = F(\|x - c_m\|, r_m) \tag{1}$$

A typical activation function can be $F(u,r) = \exp(-u/r)$. More details on the choice of the activation function can be found in [7]. Henceforth, unless otherwise

stated, it will be considered that $r_m = r = 1, m = 1, M$. Therefore, the network's response y to a given input x is:

$$y = \Psi H x \tag{2}$$

The network training is equivalent to the computation of the synaptic weights ψ_m, $m = 1, M$. It is performed by presenting the network with a number T of input–output pairs — i.e. airfoil shapes paired with cost function values. The number and the selection of the centers of the hidden units is strongly related to capacity of the network to approximate a smooth input–output mapping. The M centers should be chosen among the T input vectors and this choice affects substantially the predictive capabilities and the generalization of the network. Since the number of training patterns is relatively moderate, the choice $M = T$ can be made. If $\hat{x}^{(t)}$ ($t = 1, T$) denotes the t-th input pattern and $\hat{y}^{(t)}$ the corresponding output, then the network training reduces to the solution of the linear system:

$$\hat{H}\psi = \hat{y} \tag{3}$$

Where $\hat{y} = (\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_T)$ are the known outputs of the training patterns and matrix $\hat{H}$ contains the responses of the hidden nodes to these patterns:

$$\hat{H}_{m,t} = H_m(\hat{x}^{(t)}),\ m = 1, M \textit{ and } t = 1, T$$

In the case of $M = T$, $\hat{H}$ is a real symmetric positive-definite matrix.

A noticeable improvement over the standard *RBF* network has been proposed in [5], based on the self-adaption of the network model to the problem itself. So, in Eq. 1, a modified norm is introduced that takes into account the importance of each of the design variables with respect to the network's response, i.e. to the fitness fucntion. Practically, the standard norm-2 is replaced by its weighted variant:

$$u_m = \|x - c^{(m)}\|_{w,2} = \left(\sum_{i=1}^{N} I_i (x_i - c_i^{(m)})^2 \right)^{1/2} \tag{4}$$

Where I_i quantify the relative importace of the design variables and they are calculated as follows:

$$I_i = \frac{\left|\frac{\partial y^*}{\partial x_i}\right|}{\|\nabla y^*\|_1} \tag{5}$$

The superscript $*$ denotes computations at a characteristic point, such being the current optimum point of the genetic evolution.

The *RBF* network can provide estimations not only of the fitness function, but of its partial derivatives as well:

$$y = \sum_{m=1}^{M} \psi_m H_m \tag{6}$$

$$\frac{\partial y}{\partial x_i} = \sum_{m=1}^{M} \psi_m H'_m \alpha_{i,m} \tag{7}$$

$$\frac{\partial^2 y}{\partial x_i^2} = \sum_{m=1}^{M} \psi_m \left(H''_m \alpha_{i,m}^2 + \frac{H'_m}{u_m}(I_i - \alpha_{i,m}^2) \right) \tag{8}$$

$$\frac{\partial^2 y}{\partial x_i x_j} = \sum_{m=1}^{M} \psi_m \alpha_{i,m} \alpha_{j,m} \left(H''_m - \frac{H'_m}{u_m} \right) \tag{9}$$

where

$$H_m = H(u_m)$$
$$H_m^{(\kappa)} = \frac{d^{(\kappa)} H(u_m)}{d u_m^{(\kappa)}}$$
$$\alpha_{i,m} = I_i \frac{x_i - c_i^{(m)}}{u_m}$$

The Importance Factors, I_i, have an autocatalytic effect over the *RBF* network performance, since they are calculated by the network itself and, in turn, they affect its further trainings.

3 Model Reduction

3.1 Orthogonal Decomposition

Suppose a design point $x = (x_1, \ldots, x_N)$, in the Euclidian space $\mathcal{E}^N$, to be optimized with respect to multiple criteria $f_j : \mathcal{E}^N \to \mathbb{R},\ j = 1, M$ in an M-objective problem. An important information would be to locate the variables or, in a more general sense, the directions in the search space that affect most each criterion f_j. To do so, we first focus on the case of a single cost function and consider the corresponding single-objective problem. The cost function f — index j is omitted — can be approximated as follows:

$$f(x) = f(x_0) + \nabla f(x_0)^\top \delta x + \frac{1}{2} \delta x^\top H(x_0)\, \delta x + \cdots \tag{10}$$

Where $\delta x = x - x_0$ is the distance from a given point x_0 in the search space and

$$H(x_0) = \left(\frac{\partial^2 f}{\partial x_i \partial x_j}(x_0) \right)$$

is the Hessian matrix calculated at x_0.

We diagonalize the Hessian matrix:

$$H(x_0) = \Phi \Lambda \Phi^\top \tag{11}$$

Since H is real symmetric:

$$\Lambda = diag(\lambda_1, \lambda_2, \ldots, \lambda_N), \quad \lambda_i \in \mathbb{R}$$

Given that δx is transformed to $\delta \xi = \Phi^\top \delta x$, the terms in Eq. 10 can be expressed as follows in the basis of the eigenvectors of H:

$$\delta x^\top H(x_0)\, \delta x = \delta \xi^\top \Lambda\, \delta \xi = \sum_{i=1}^{N} \lambda_i\, \delta \xi_i^2 \tag{12}$$

$$\nabla f(x_0)^\top \delta x = (\Phi^\top \nabla f(x_0))^\top \delta \xi = \sum_{i=1}^{N} \alpha_i\, \delta \xi_i \tag{13}$$

Where $\alpha_i = [\Phi^\top \nabla f(x_0)]_i$ are the components of the gradient of f expressed in the new basis.

Therefore, combining Eq. 10, 12 and 13, the approximation of f in the basis of eigenvectors of the Hessian matrix simplifies to the following:

$$f(x) = f(x_0) + \sum_{i=1}^{N} (\alpha_i \delta\xi_i + \frac{1}{2}\lambda_i \delta\xi_i^2) \tag{14}$$

A necessary condition for an extremum in the above function is $\frac{\partial f}{\partial \xi_i} = 0$, which leads to the following values of $\delta\xi_i$:

$$\alpha_i + \lambda_i \delta\xi_i = 0 \Leftrightarrow \delta\xi i^* = -\frac{\alpha_i}{\lambda_i} \tag{15}$$

Each new variable $\delta\xi_i$ contributes to the value of f by a certain amount $\delta f_i = \delta f_i(\delta\xi_i)$. From the above values of $\delta\xi_i^*$ we get:

$$\delta f_i^* = \alpha_i \delta\xi i^* + \frac{1}{2}\lambda_i \delta\xi_i^{*2} = \frac{-1/2\alpha_i^2}{\lambda_i} \tag{16}$$

The variables with the greatest pay-off, with respect to the extremum, are those that cause the greatest variation δf_i to f. Thus, the most paying directions in the search space are those associated with the largest values of the quantity:

$$\left| \frac{\alpha_i^2}{\lambda_i} \right| = \frac{[\Phi \nabla f(x_0)]_i^2}{|\lambda_i|} = \zeta_i \tag{17}$$

By ordering ζ_i, such that $\zeta_1 \geq \cdots \geq \zeta_p \geq \cdots \geq \zeta_N$, optimization can be restricted to the subspace $\mathcal{E}^p$, such that $\mathcal{E}^N = \mathcal{E}^P \oplus \mathcal{E}^{N-P}$, with the greatest pay-off, achieving, thus, faster convergence. In a multi-objective problem, the above analysis would be performed separately for each criterion f_j, aiming at the identification of a "best subspace" for each of them.

3.2 Practicalities

A few practical issues arise when implementing the present model reduction technique. They are discussed in the following paragraphs.

3.2.1 Search field mapping

The field, where solutions are searched by a *GA*, is normally a convex set of the form $[x_{1,min}, x_{1,max}] \times [x_{2,min}, x_{2,max}] \times \ldots \times [x_{N,min}, x_{N,max}]$. The exact mapping of this rectangle in the new space requires 2^N computations. To avoid this extremely costly approach, the following is proposed:

1. The initial search intervals are normalized, so as their extent to be equal to 1.
2. The radius of the circle that circumsribes the normalized rectangle, thus square, is $r = \frac{1}{2}\sqrt{N}$. The new search field is then $[c-r, c+r]^N$, where c is the barycenter of the normalized initial search field, mapped in the new basis.

The above approach ensures that the initial search field is a subset of the new one and, hence, all possible solutions are retained to consideration. However, this inflation deteriorates the exploration ability of the algorithm. To alleviate this side-effect a shrinked radius, $\rho = \kappa r$, can be used instead, where $\kappa = 0.5 \sim 0.7$.

A counterpart of this size reduction is that certain variables of at least one individual, after being mapped into the new space, may happen to fall outside the new interval. In this case, the search field is enlarged in the direction of these variables only, by $\delta\rho = \lambda r$, where $\lambda = 0.1 \sim 0.2$.

3.2.2 Dimension of Reduced Space

Having ordered ζ_i, derived from Eq. 17, an issue that directly arises is how many of the corresponding eigenvectors to keep for the reduced space optimization. An approach, which seems to perform well, is to specify a minimum dimension d of the reduced space and additionally keep the eigenvectors v_i, for which the corresponding value ζ_i is of the same order of magnitude as ζ_d. Formally, this can be expressed as:

$$\mathcal{E}^P = \text{lin}\{v_i : \text{ceil}(\log(\zeta_i)) \geq \text{rint}(\log(\zeta_d))\} \tag{18}$$

where lin denotes the *linear span* of a set, rint() rounds its argument to the closest integral value and ceil() computes the smallest integral value not less than its argument (ceiling). The use of ceil() instead of rint() intends to counterbalance eventual underestimation of the importance of a direction in the search space, due to the inevitable inaccuracy of the surrogate model. In addition, the adequacy of the filtering implied by Eq. 18 can be further adjusted by the selection of the logarithmic base.

4 The Algorithm

The aim of the developed algorithm is to exploit the results of the orthogonal decomposition of the cost function, described in the previous section, and restrict optimization in the most paying search subspace. Hence, a better exploration can be performed resulting in faster convergence. To avoid being trapped in local optima, partial and full optimization are regularly alternated. In addition, this alternation allows a continuous adaption of the optimum subspace.

The proposed algorithm is outlined hereafter:

Phase 1: The starting population keeps evolving for a few generations. The genetic operators apply on all the design variables. The evaluated individuals are kept in a database, along with their cost or fitness function values.

Phase 2: An *RBF* network is trained locally around the current best solution. As soon as the relative error of the network's prediction $\tilde{f}$ with respect to the exact value f of the fitness function, i.e. $\left|f-\tilde{f}\right|/|f|$, is less than a threshold ε, the model reduction is performed. This reduction involves the following steps:

(a) Mapping of the population in the new basis.

(b) Mapping of the search field in the new basis.

(c) A small percentage, $\sigma < 5\%$, of the population is moved to the region of the local optimum to enhance exploration.

The population keeps evolving for a number of generations, with the genetic operators being applied only to the variables identified as the most important.

Phase 3: The *GA* shifts to full optimization. To avoid eventual premature convergence, the variables that were exempted from the genetic operators are mutated with a high probability $p \simeq 90\%$. Only a randomly chosen part ($20 \sim 30\%$) of the population is submitted to this forced mutation. The population keeps evolving for a number of generations, with the genetic operators being applied to all the design variables.

Phases 2 and 3 are alternated up to convergence. Whenever the quality of the *RBF* network is acceptable, the new basis, which will be used for the next model reduction, is updated. Throughout the genetic evolution, the database is enriched with the newly examined individuals.

5 Method Evaluation

The test-cases used for the evaluation of the proposed method intend to show that the self-adaption of the search space reduces substantially the computational cost, comparing to the conventional *GA*. In the latter, throughout the evolution process, genetic operators apply on all the design variables.

Prior to airfoil design problems the algorithm was tested with a numerical case. The *Ackley* function has been selected, a typical multi-modal function, well-suited for the assessment of optimization methods.

For the first two inverse design problems, a simple flow model, namely the panel method [9] for incompressible, irrotational airfoil flow was used. This evaluation tool is very fast, and this permitting a high number of tests to be made, the results presented are typical of several computations. Thus, despite the stochastic nature of *GAs*, conclusions admit of generalization.

For the last two problems a time-marching solver for the compressible Euler equations was used (see [10], [11]). A cell-centered finite-volume scheme is used on a structured quadrangular mesh. The evaluation of a candidate solution involves (a) the generation of the profile contour from the design variables, (b) the solution of the flow equations and (c) the post-processing of the results, so as to extract the cost function value.

All of the test-cases have been formulated as minimization problems. To facilitate notation, the *GA* that employs the Model Reduction will be denoted as *GA-RM*. The basic parameters related to the genetic operators that have been used in all the test-cases are listed in Table 1. Typical values for the configuration of the *GA-RM* scheme are given in Table 2 (refer to Section 4). The test-cases involving the solution of Euler equations were evaluated using the PVM [12] interface on a Linux cluster, and, thus, the computational time was substantially reduced (less than 25 min. on 30 processors for 2000 evaluations).

Population size	30
Two-point crossover probability	85%
Mutation probability	0.5%
Binary tournament probability	85%
Coding type	Gray binary

Table 1: Basic *GA* parameters

Generations using the Reduced Model	$8 \sim 12$
Generations using the Full Model	$3 \sim 5$
Percentage of the population forcibly mutated [1]	25%
Forced mutation probability [2]	90%

Table 2: Typical *GA-RM* configuration

5.1 Minimization of the *Ackley* function

The problem consists in the minimization of the function:

$$f(x) = e + 20\left[1 - \exp\left(-0.2\sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}\right)\right] - \exp\left[\frac{1}{N}\sum_{i=1}^{N}\cos(2\pi x_i)\right] \qquad (19)$$

The location of the minimum is the $\{0\}$. N was set to 14 and the search interval for all the variables to $[-3,3]$. The landscape of the *Ackley* function for 2 variables is illustrated in Fig. 2.

From Eq. 19 it is obvious that the function is axisymmetric and hence there is no privileged direction in the search space. However, the *GA-RM* scheme outperforms the conventional *GA*, as demonstrated in Fig. 3. The reason is that as the optimization process evolves, locally, some of the variables have a greater impact on the function value, since the rest of them is already set to a locally optimum value. The continuous adaption of the Reduced Model circumvents the "traps" of local optima. In this case, the minimum dimension of the Reduced Model was 7.

[1] When shifting from the Reduced to the Full Model optimization.

[2] This concerns the variables exempted from the Reduced Model.

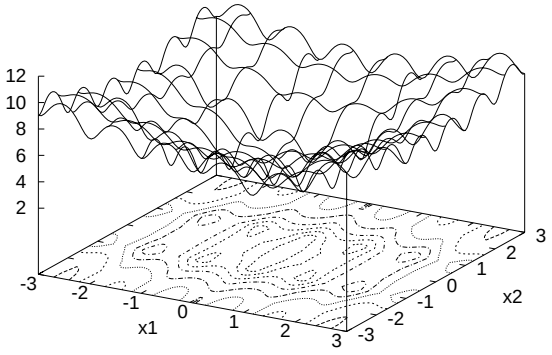

Figure 2: The *Ackley* function.

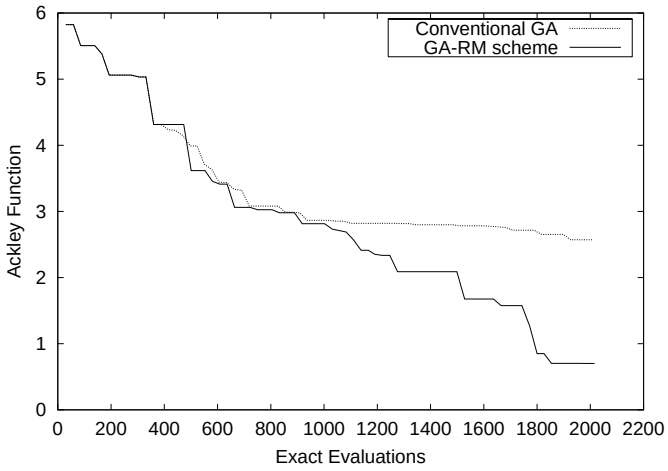

Figure 3: *Ackley* function: Convergence history.

5.2 Reconstruction of the *NACA0012* profile

In the first study, the aim was to compute the airfoil shape that yields a given pressure distribution. As target the pressure distribution of the well-known *NACA0012* profile at zero incidence was set, calculated by the panel method.

The airfoil parameterization uses the Bezier-Bernstein polynomials. Two curves, sharing their first and last control point, were used for the parameterization of the pressure (*PS*) and the suction side (SS) separately. Design parameters are the coordinates of the control points between the leading (*LE*) and the trailing (*TE*) edge. Usually, to ensure a rounded *LE*, the abscissas of the first free control points are set equal to that of the *LE*. This parameterization will be used in all airfoil design cases, unless otherwise stated.

In this case, 4 free control points for each side were used ($PS_i, SS_i,\ i = 1,4$), consequently $2 \cdot (1 \cdot 1 + (4-1) \cdot 2) = 14$ design variables (Table 3).

1	2	3	4	5	6	7
$y(PS_1)$	$x(PS_2)$	$y(PS_2)$	$x(PS_3)$	$y(PS_3)$	$x(PS_4)$	$y(PS_4)$
8	9	10	11	12	13	14
$y(SS_1)$	$x(SS_2)$	$y(SS_2)$	$x(SS_3)$	$y(SS_3)$	$x(SS_4)$	$y(SS_4)$

Table 3: Design variables notation

The convergence of the *GA-RM* scheme, in comparison with the conventional *GA*, is illustrated in Fig. 4. In this case, there is no significant improvement in the performance of the algorithm. The reason can be understood from Fig. 5. In the course of the optimization process, no particular variable was found dominant throughout the entire convergence process, but only for one time. Therefore, the exploration of the search space was not significantly accelerated, since no significant model reduction could be performed.

5.3 Reconstruction of the *NACA4412* profile

The second problem is similar to the first one, but with a non-symmetric airfoil. This study aims at the reconstruction of the *NACA4412* profile at incompressible, irrotational flow conditions and 10^o of incidence. As in the previous case, 14 design parameters were used and the pressure coefficient distribution along the airfoil walls

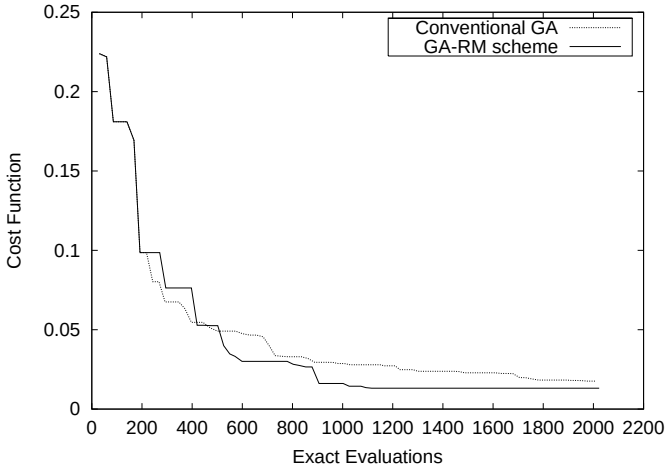

Figure 4: *NACA0012*: Convergence history.

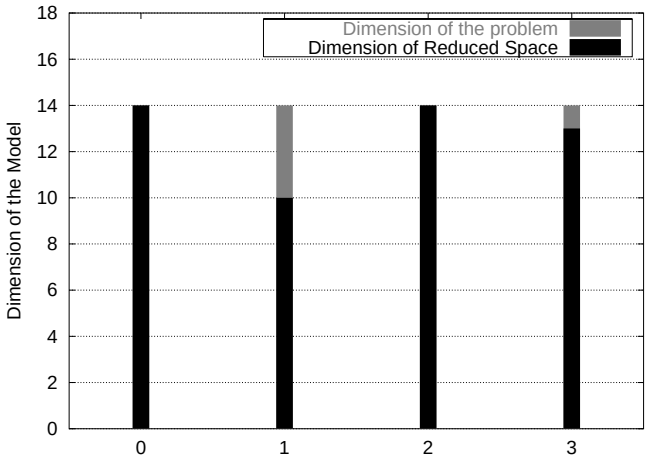

Figure 5: *NACA0012*: Dimension of the Reduced Space at different instants.

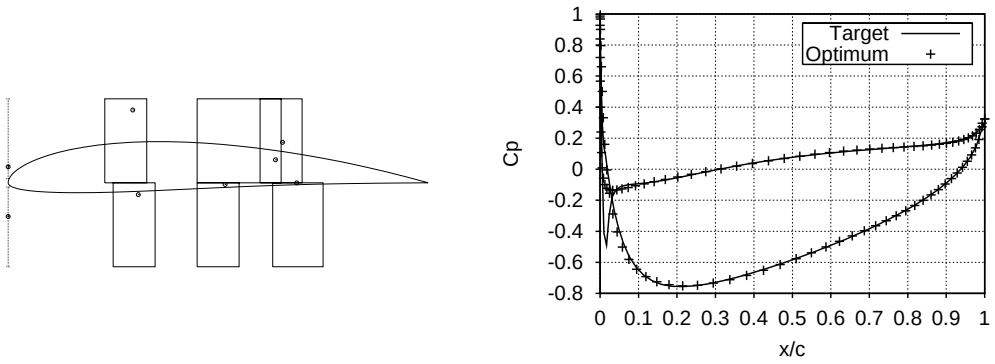

Figure 6: *NACA4412*:
Left: Best computed profile, optimum location of Bezier control points and search fields.
Right: Target and best computed pressure coefficient distribution along the airfoil contour.

was computed with the panel method. The best airfoil shape captured by the *GA-RM* algorithm, along with the optimum location of the Bezier control points and the corresponding search fields, are shown in Fig. 6.

In this test case, the auto-adaptive Model Reduction results in a substantially improved performace, compared to the conventional *GA* (Fig. 7). It is important to mention that the faster convergence of the *GA-RM* scheme is obtained within the same number of generations — apart from the same computational cost — as that of the conventional *GA*. The reason can be deduced from Fig. 8, where it can be seen that a considerable reduction in the dimension of the search space has been performed at different stages of the genetic evolution.

5.4 Inverse design of a High-Lift profile

The High-Lift profile, which was set as target in this inverse design, as well as the Low-Drag one, in the next subsection, have been defined and tested in [13], [14] and [15]. As already mentioned in Section 5, the two-dimensional Euler equations are solved for the evaluation of a candidate solution. A typical structured mesh, generated by conformal mapping techniques in curvilinear coordinates, is shown in Fig. 9. In this test-case only, the profile parameterization corresponds to a linear

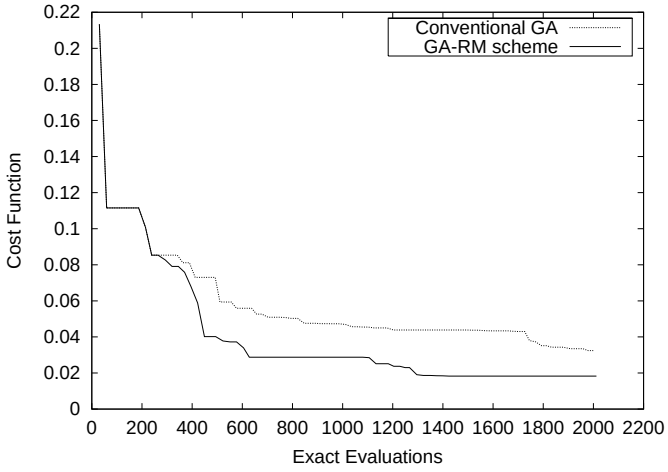

Figure 7: *NACA4412*: Convergence history.

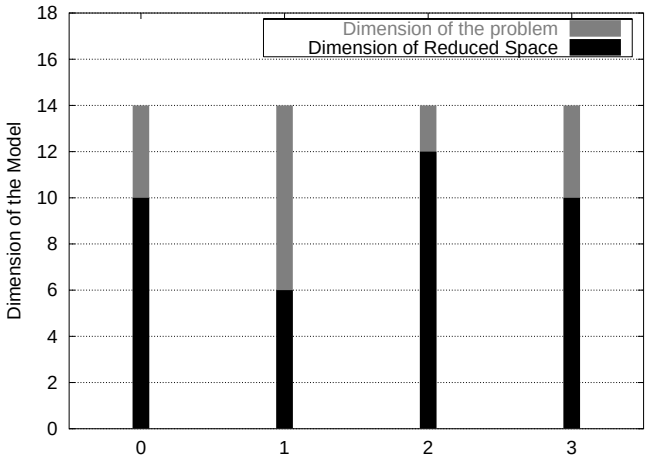

Figure 8: *NACA4412*: Dimension of the Reduced Space at different instants.

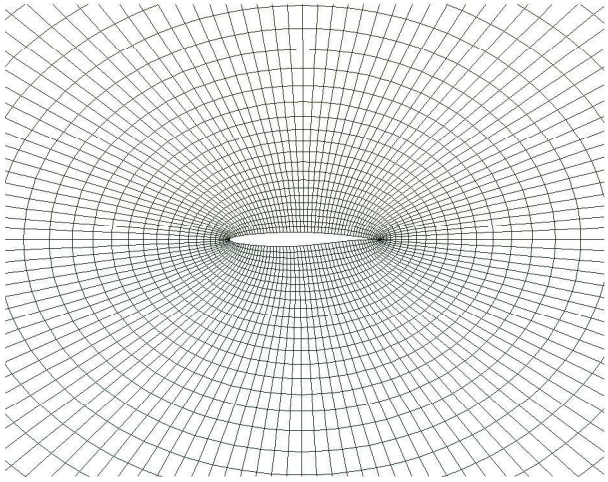

Figure 9: Partial view of an *O-type* mesh.

combination of 14 Hicks-Henne shape functions, [16], for each airfoil side. In brief, a profile $y(x)$ can be obtained by the following summation:

$$y(x) = \sum_{i=1}^{N} b_i f_i(x) \tag{20}$$

where f_i are the basis functions:

$$f_i(x) = \sin^{\alpha_i}\left(\pi x^{\beta_i}\right)$$

in which α_i and β_i are a priori adjusted to convenient values (see Table 4 and [14]). In particular, β_i are related to the peak locations, p_i of the corresponding shape functions f_i through the formula:

$$\beta_i = \frac{\ln(0.5)}{\ln(p_i)}$$

Therefore, the parameters to be optimized are the coefficients b_i in Eq. 20. The *PS* and the SS, as already mentioned, are parameterized separately, resulting, thus, in 28 design variables.

The target pressure distribution, at freestream conditions of $M_\infty = 0.2$ and $a = 10.8^o$, along with the pressure distribution of the best profile, as computed by the

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14
p_i	.025	.05	.1	.15	.2	.3	.4	.5	.6	.7	.8	. 85	.9	.95
α_i	2.	2.	2.	3.	3.	3.	3.	3.	3.	3.	3.	2.	2.	1.

Table 4: Adjustment of Hicks-Henne parameterization

GA-RM scheme, are plotted in Fig. 10. In Fig. 11 the convergence of the *GA-RM* algorithm is compared to that of the conventional *GA*.

5.5 Inverse design of a Low-Drag profile

This test-case is similar to the previous one, consisting in the reconstruction of a Low-Drag profile from the pressure distribution, as calculated by the compressible Euler equations. However, the Bezier parameterization has been used in this case, as described in Section 5.2. 8 free control points have been chosen for each side of the airfoil, resulting, thus, in 30 design variables.

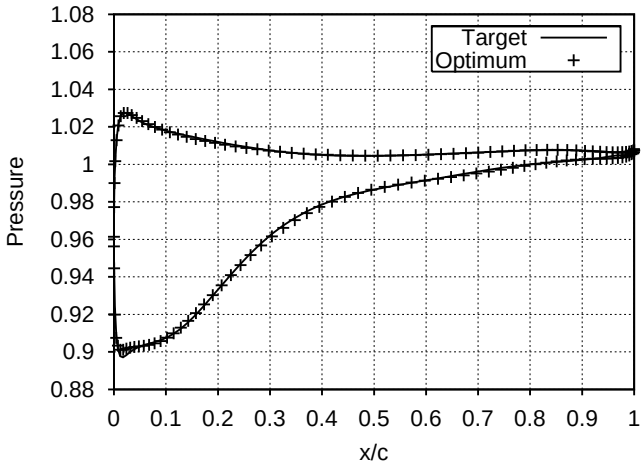

Figure 10: High-Lift profile: Target and best computed pressure $((p - p_\infty)/p_\infty)$ distribution along the airfoil contour.

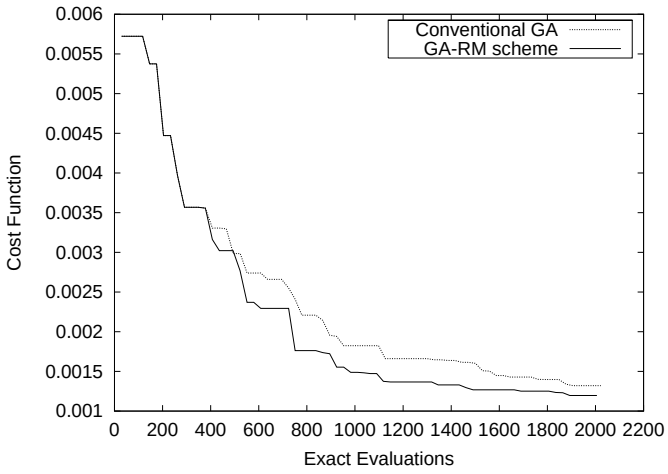

Figure 11: High-Lift profile: Convergence history.

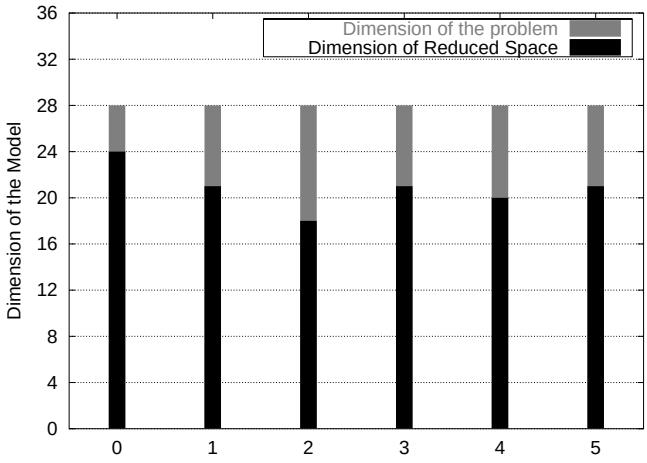

Figure 12: High-Lift profile: Dimension of the Reduced Space at different instants.

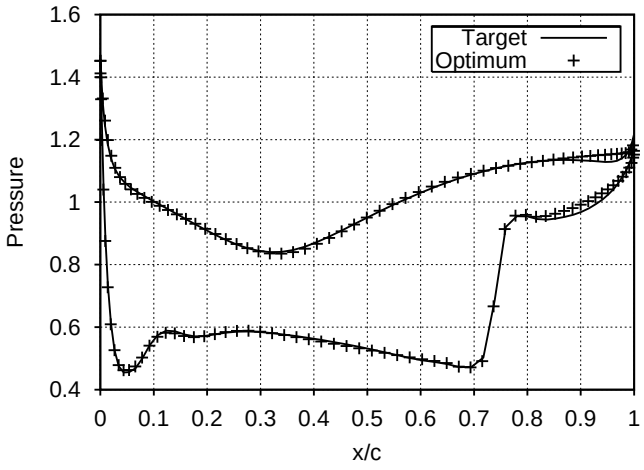

Figure 13: Low-Drag profile: Target and best computed pressure $((p - p_\infty / p_\infty)$ distribution along the airfoil contour.

The freestream conditions were: $M_\infty = 0.77$ and $a = 1.0^o$ — representing a typical cruise regime. The target pressure distribution at these conditions and the best one obtained by the *GA-RM* scheme are plotted in Fig. 13.

Using the Bezier parameterization, a considerable reduction in the dimension of the search space could be performed during the optimization process (compare Fig. 12 and Fig. 15). The result was a singificant improvement in the performance of the *GA-RM* algorithm (Fig. 14), despite the higher number of design parameters, compared to the previous problem. In fact, the quality of the solution, achieved by the conventional *GA* after 3000 evaluations, is attained by the *GA-RM* scheme with only 1000 evaluations.

The considerable reduction of the search space dimension is in conformity with the general observation that, in aerodynamic shape-optimization problems, the ordinates of the Bezier control points tend to affect more the quality of a solution. This non-uniform "importance" permitted to periodically restrict optimization to the most "important" sub-space and, thus, the acceleration of convergence.

In Fig. 16 it is illustrated the application of Eq. 18, concerning the selection of the Reduced Model dimension at a given moment of the optimization process. For each direction in the search space, the corresponding values $\log(\zeta_i)$ are plotted, as

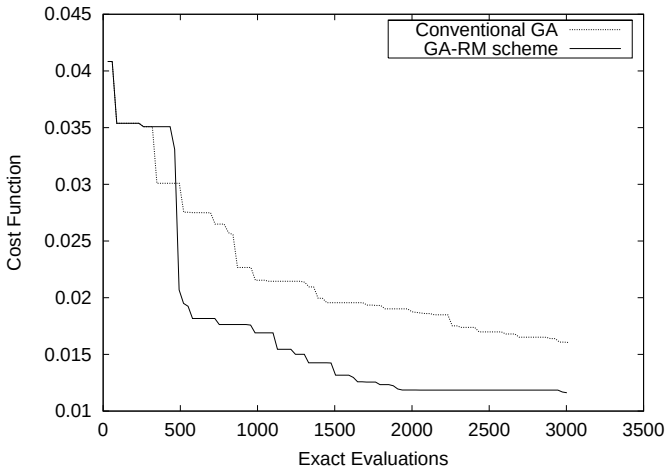

Figure 14: Low-Drag profile: Convergence history.

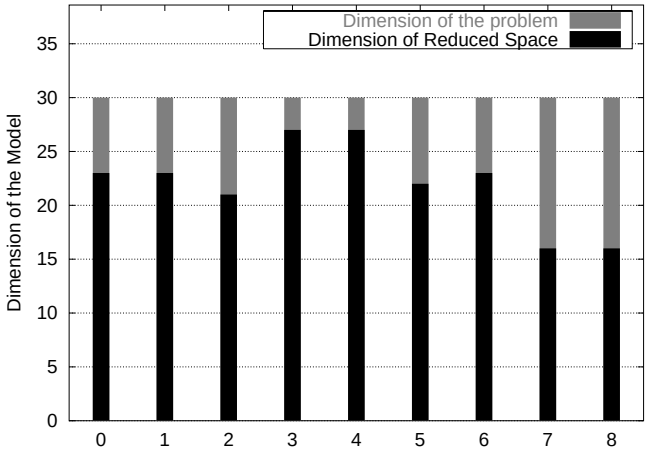

Figure 15: Low-Drag profile: Dimension of the Reduced Space at different instants.

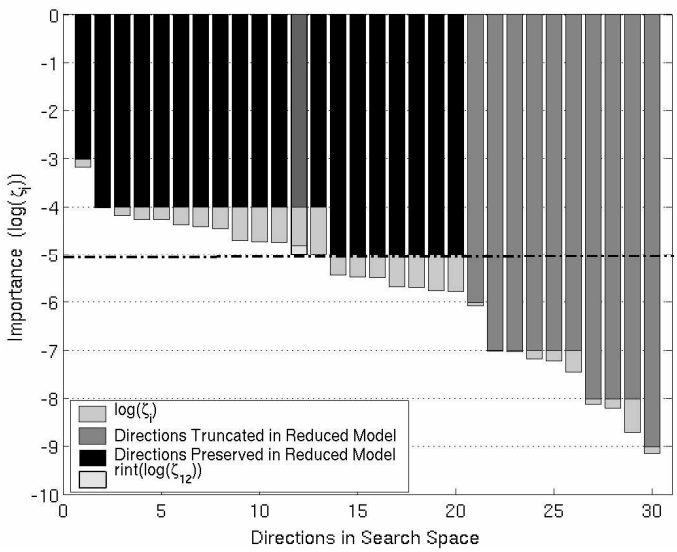

Figure 16: Low-Drag profile: Selecting the Reduced Model dimension.

well as their ceilings, which are used to determine the dimension of the reduced space. The minimum dimension d of the Reduced Model was set to be 40% of the dimension of the optimization problem, thus $d = 12$. Therefore, the reference value, used to identify which directions are to be ignored in the reduced space, was taken to be $\mathrm{rint}(\log(\zeta_{12}))$. This value, plotted with a different colour, indicates the directions to be kept in the Reduced Model.

6 Conclusions

In this report, a self-adaption of the search space to the optimization problem itself has been proposed and tested with the objective to reduce the optimization cost — a particularly important parameter in *CFD* design cases. This adaption involves the following steps:

(a) Mapping of the initial search-space into the basis of the Hessian matrix eigenvectors. The directions of the new basis take into account possible interactions between the initial design variables.

(b) Adaption of the search field in the new basis, taking into consideration the dispersion of the population.

(c) Truncation of the less important directions of the mapped space.

The reduction of the search-space dimension can substantially limit the number of generations, and thus of costly evaluations of candidate solutions, needed to achieve a given level of design quality. However, if the formulation or the nature of an optimization problem attribute a quite uniform importance to all the design variables, then no significant model reduction can be performed. In the latter case, the adaption of the optimization process is, obviously, less effective.

The role of *ANNs*, as a substitute to the costly *CFD* evaluation tools, is of primordial importance, since all the mathematical analysis is based on their approximations. In particular, the *RBF*networks, used in this study, seem to perform satisfactorily, combining a low training cost. However, other types of *ANN* should also be tested, even of higher training cost, since the Model Reduction is a subtle operation.

The present method is currently being extended to determine the most "paying" sub-space in a multi-objective optimization formulation. In this case, identifying which directions of the search space affect most each criterion, is the element of information to exploit.

References

[1] D.E. GOLDBERG. *Genetic Algorithms in search, optimization & machine learning*. Addison-Wesley, 1989.

[2] Z. MICHALEWICZ. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, Berlin Heidelberg, 2nd edition, 1994.

[3] K.C. GIANNAKOGLOU. Acceleration of Genetic Algorithms Using Artificial Neural Networks - Theoretical Background. Von-Karman Institute LS 2000-07, May 2000.

[4] K.C. GIANNAKOGLOU and A.P. GIOTIS. Acceleration of Genetic Algorithms Using Artificial Neural Networks - Application of the Method. Von-Karman Institute LS 2000-07, May 2000.

[5] K.C. GIANNAKOGLOU, A.P. GIOTIS, and M.K. KARAKASIS. Low–cost Genetic Optimization Based on Inexact Pre–Evaluation and the Sensitivity Analysis of Design Parameters. *Inverse Problems in Engineering*, 9:389–412, 2001.

[6] T. POGGIO and F. GIROSI. Networks For Approximation and Learning. *Proceedings of The IEEE*, 78(9):1481–1497, 1990.

[7] A. CICHOCKI and R. UNBENHAUEN. *Neural Networks For Optimization and Signal Processing*. John Wiley & Sons, 1993.

[8] S. HAYKIN. *Neural Networks*. Prentice Hall International, Inc., 2nd edition, 1998.

[9] V. DEDOUSSIS. Calculation of Inviscid Flow through Aerofoil Cascade and between Wind Tunnel Wall. Master's thesis, Dept. of Aeronautics, Imperial College, London, 1983.

[10] A. JAMESON, W. SCHMIDT, and E. TURKEL. Numerical Solution of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time Stepping Schemes. In *AIAA 14th Fluid Dynamics and Plasma Dynamics Conference, Palo Alto*, number 81-1259 in AIAA Papers, 1981.

[11] T.H. PULLIAM and J.L. STEGER. Recent Improvements in Efficiency, Accuracy and Convergence for Implicit Approximate Factorization Algorithms. In *AIAA 23rd Aerospace Sciences Meeting, Reno*, number 85-0360 in AIAA Papers, January 1985.

[12] A. GEIST, A. BEGUELIN, J. DONGARRA, W. JIANG, R. MANCHEK, and V. SUNDERAM. *PVM: Parallel Virtual Machine. A user's guide and tutorial for networked Parallel Computing*, 1994.

[13] J. PERIAUX, G. BUGEDA, P.K. CHAVIAROPOULOS, K. GIANNAKOGLOU, S. LANTERI, and B. MANTEL, editors. *Optimum Aerodynamic Design & Parallel Navier-Stokes Computations. ECARP – European Computational Aerodynamics Research Project.*, volume 61 of *Notes on Numerical Fluid Mechanics*. Community Research in Aeronautics, 1998.

[14] Z.-L. TANG, J.-A. DESIDERI, and J. PERIAUX. Multi-Objective Optimization and Inverse Problems using Control Theory and Nash Games. In *ECCOMAS CFD2001 Conference*, September 2001.

[15] Z.-L. TANG, J.-A. DESIDERI, and J. PERIAUX. Distributed Optimization using Virtual and Real Game Strategies for Aerodynamic Design. INRIA, 2002. Research report.

[16] R.M. HICKS and P.A. HENNE. Wing design by numerical optimization. *Journal of Aircraft*, 15(7):407–412, 1978.

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifi que
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38330 Montbonnot-St-Martin (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399